

### **REMARKS**

Claims 1-17 are pending and under consideration. Claims 1, 2, 3, 5, 6, 7, 9, 10, 11, 13, 14, 15, and 17 are amended herein. Support for the amendments to the claims may be found in the claims as filed originally, in Fig. 7, and in the specification at page 7, lines 7-12.

Reconsideration is requested based on the foregoing amendments and the following remarks.

#### **Response to Arguments:**

The Applicants appreciate the consideration given to their arguments, and the new grounds of rejection. Further consideration is requested.

#### **Claim Rejections – 35 U.S.C. § 103:**

Claims 1, 2, 4, 5, 6, 8, 9, 10, 12, 13, 14, 16, and 17 are rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,117,186 to Wydall et al. (hereinafter "Wydall") in view of U.S. Patent No. 6,738,966 to Tanaka (hereinafter "Tanaka"). The rejection is traversed to the extent it might apply to the claims as amended.

The second clause of claim 1 recites:

A first storage medium readable by a computer and storing a script containing parameters.

Neither Wydall nor Tanaka teach, disclose, or suggest "a first storage medium readable by a computer and storing a script containing parameters," as recited in claim 1. Wydall, rather, describes storing video parameters in a file specially and previously created by the CD-ROM program (e.g., cdnav.ini) or in a line in the WIN.INI file. In particular, as described at column 6, lines 52, 53, and 54:

The video parameters may be stored in a file specially and previously created by the CD-ROM program (e.g., cdnav.ini) or in a line in the WIN.INI file.

Since, in Wydall, video parameters are stored in a file specially and previously created by the CD-ROM program or in a line in the WIN.INI file, Wydall has no "first storage medium readable by a computer and storing a script containing parameters," as recited in claim 1.

Tanaka, for its part, relates to a compiling device translating a source program into an object program such as a machine language program or an assembler program. In particular, is described at column 1, lines 13-18:

The present invention relates to a compiling device translating a source program into an object program such as a machine language program or an assembler program, and in particular, to improvements achieved when the source program includes a section written in high-level programming language and a section written in assembly language.

Since Tanaka relates to a compiling device translating a source program into an object program such as a machine language program or an assembler program, Tanaka has no "first storage medium readable by a computer and storing a script containing parameters" either, and thus cannot make up for the deficiencies of Wydall with respect to claim 1.

An object of the claimed invention is to configure a system by executing steps specified by a script. The system is configured in a way that replaces the parameter of the script with a value of the parameter itself. The fifth clause of claim 1, in particular, recites:

A module executing steps specified by the script in a way that replaces a parameter of the script with a value of the parameter.

Wydall neither teaches, discloses, nor suggests "executing steps specified by the script in a way that replaces a parameter of the script with a value of the parameter," as recited in claim 1. In Wydall, rather, the optimum size and/or frame rate is determined for this computer 2 by examining the operation of the computer 2 during playback of the video, not by "executing steps specified by the script in a way that replaces a parameter of the script with a value of the parameter," as recited in claim 1. In particular, as described at column 6, lines 60-67, continuing at column 7, lines 1-3:

If the CD-ROM program has not previously been used on this computer 2 (or if the CD-ROM program has been "uninstalled", e.g., the video parameters have been deleted from the hard disk 8), then a test video is loaded and played at full screen size (step 216). By examining the operation of the computer 2 during playback of the video, the optimum size and/or frame rate is determined for this computer 2 (step 218). This information is saved on the hard disk 8, e.g., in a cdnav.ini or the WIN.INI file, as discussed above (step 220) and the video is played back according to these parameters.

Since, in Wydall, the optimum size and/or frame rate is determined for this computer 2 by examining the operation of the computer 2 during playback of the video, Wydall is not "executing steps specified by the script in a way that replaces a parameter of the script with a value of the parameter," as recited in claim 1.

The Office Action notes this deficiency of Wydall, and attempts to compensate for it by

modifying Wydall in view of Tanaka. Tanaka, however, is not “executing steps specified by the script in a way that replaces a parameter of the script with a value of the parameter,” as recited in claim 1, either, and thus cannot make up for the deficiencies of Wydall with respect to claim 1. In Tanaka, rather, “formal parameters are replaced with variables generated inside a compiler,” as noted in the Office Action at page 3, line 10.

Tanaka, in fact, is allocating variables to registers that differ from the registers updated during the inline assembly subroutine, not replacing “a parameter of the script with a value of the parameter,” as recited in claim 1. In particular, as described in the Abstract:

Then, if a variable allocated to a register has a live range (lifetime) that includes an inline assembly subroutine, the variable is allocated a register that differs from the registers updated during the inline assembly subroutine.

Since Tanaka is allocating variables to registers that differ from the registers updated during the inline assembly subroutine, Tanaka is not “executing steps specified by the script in a way that replaces a parameter of the script with a value of the parameter,” as recited in claim 1.

In Tanaka, moreover “a register parameter replacing unit... replaces all of the formal parameter values in the functions with the temporary variables indicated by the substitution instructions,” as noted in the Office Action at page 3, lines 15 and 16, rather than replacing “a parameter of the script with a value of the parameter,” as recited in claim 1. In particular, as described in Tanaka at column 4, lines 4-13:

Here, the compiling device further includes a register parameter replacing unit that, when formal parameters that should use registers exist, (1) generates substitution instructions for substituting temporary variables for the values of each formal parameter used in the functions, and inserts each of the generated substitution instructions at the start of a corresponding function, and (2) replaces all of the formal parameter values in the functions with the temporary variables indicated by the substitution instructions.

Since, in Tanaka, a register parameter replacing unit replaces all of the formal parameter values in the functions with the temporary variables indicated by the substitution instructions, Tanaka is not “executing steps specified by the script in a way that replaces a parameter of the script with a value of the parameter,” as recited in claim 1.

Furthermore, in Tanaka, if it is desirable to insert instructions capable of performing both division and remainder calculation at a plurality of places in the program, a *macro* is defined as in FIG. 3A, rather than replacing “a parameter of the script with a value of the parameter,” as

recited in claim 1. In particular, as described in Tanaka at column 2, lines 28-33:

Furthermore, if it is desirable to insert instructions capable of performing both division and remainder calculation at a plurality of places in the program, a macro is defined as in FIG. 3A, and if this macro is used as shown in FIG. 3B efficiency is increased.

Since, in Tanaka, if it is desirable to insert instructions capable of performing both division and remainder calculation at a plurality of places in the program, a macro is defined as in FIG. 3A, Tanaka is not “executing steps specified by the script in a way that replaces a parameter of the script with a value of the parameter,” as recited in claim 1.

Tanaka, moreover, checks the object program generated by the compiler to determine whether the value of the variable x has been destroyed. In particular, as described at column 2, lines 40-55:

The program includes a plurality of variables. When a value of a certain variable x is valid for an entire inline assembly routine, the first check operation involves thoroughly checking the object program generated by the compiler to determine whether the value of the variable x has been destroyed. A register r is allocated to the variable x by a process performed by the compiler, so that if the inline assembly routine defines the register r, the value of the register r will differ before and after the inline assembly routine.

Since, in Tanaka, the object program generated by the compiler is checked to determine whether the value of the variable x has been destroyed, Tanaka is not “executing steps specified by the script in a way that replaces a parameter of the script with a value of the parameter,” as recited in claim 1.

Tanaka, moreover, the register parameter replacing unit 16 inserts, at the start of a function, an intermediate instruction storing, as a temporary variable, a value of a formal parameter passed to the register. In particular, as described at column 8, lines 18-23:

The register parameter replacing unit 16 is activated if register parameters are included in the program that is to be translated. The register parameter replacing unit 16 inserts, at the start of a function, an intermediate instruction storing, as a temporary variable, a value of a formal parameter passed to the register.

Since, in Tanaka, the register parameter replacing unit 16 inserts, at the start of a function, an intermediate instruction storing, as a temporary variable, a value of a formal parameter passed to the register, Tanaka is not “executing steps specified by the script in a way that replaces a parameter of the script with a value of the parameter,” as recited in claim 1.

Finally, in Tanaka, the register parameter replacing unit 16 replaces the sections of the intermediate program in which the formal parameter is used with the temporary variables. In particular, as described at column 8, lines 23-26:

Further to this, the register parameter replacing unit 16 replaces the sections of the intermediate program in which the formal parameter is used with the temporary variables.

Since, in Tanaka, the register parameter replacing unit 16 replaces the sections of the intermediate program in which the formal parameter is used with the temporary variables, Tanaka is not "executing steps specified by the script in a way that replaces a parameter of the script with a value of the parameter," as recited in claim 1.

The Office Action, in any case, asserts at page 3, line 17-22, continuing at page 4, line 1, that:

Therefore, it would have been obvious to one having ordinary skill in the art to have the module executing steps that replaces a parameter of the step information with a value of the parameter, as is in the case in Tanaka, in combination with the information processing, as taught by Wydall, because the combination would have provided an improve system having configuration tools that may be used to apply changed configuration file to the component directly (i.e. without restarting or re-initializing the component ).

Neither Wydall nor Tanaka, however, claim to provide configuration tools that may be used to apply changed configuration file to the component directly (i.e. without restarting or re-initializing the component). Wydall, rather, wants to allows non-technical people to easily use CD-ROM computer programs without undertaking an installation process, while the object of Tanaka is to provide a compiling device to translate a program. It is submitted, therefore, that persons of ordinary skill in the art at the time the invention was made would not have modified Wydall as proposed in the Office Action, since neither Wydall nor Tanaka *claim* to provide the benefit attributed to them in the Office Action. Claim 1 is thus submitted to be allowable. Withdrawal of the rejection of claim 1 is earnestly solicited.

Claims 2 and 4 depend from claim 1 and add additional distinguishing elements. Claims 2 and 4 are thus also submitted to be allowable. Withdrawal of the rejection of claims 2 and 4 is earnestly solicited.

Claims 5, 6, 8, 9, 10, 12, 13, 14, and 16:

The second clauses of claims 5, 9, and 13 recite substantially:

Referring to a script containing parameters.

Neither Wydall nor Tanaka teach, disclose, or suggest "referring to a script containing parameters," as discussed above with respect to the rejection of claim 1.

The fifth clauses of claims 5, 9, and 13 recite substantially:

Executing steps specified by the script in a way that replaces a parameter of the script with a value of the parameter.

Neither Wydall nor Tanaka teach, disclose, or suggest "executing steps specified by the script in a way that replaces a parameter of the script with a value of the parameter," as discussed above with respect to the rejection of claim 1. Claims 5, 9, and 13 are submitted to be allowable as well, for at least those reasons discussed above with respect to the rejection of claim 1.

Withdrawal of the rejection of claims 5, 9, and 13 is earnestly solicited.

Claims 6, 8, 10, 12, 14, and 16 depend from claim 5, claim 9, or claim 13 and add additional distinguishing elements. Claims 6, 8, 10, 12, 14, and 16 are thus also submitted to be allowable. Withdrawal of the rejection of claims 6, 8, 10, 12, 14, and 16 is earnestly solicited.

Claim 17:

The second clause of claim 17 recites:

Checking values of parameters contained in a script.

Neither Wydall nor Tanaka teach, disclose, or suggest, "checking values of parameters contained in a script," as discussed above with respect to the rejection of claim 1.

The fourth clause of claim 17 recites:

Replacing a parameter of the step information with a value of the parameter by executing steps specified by the script.

Neither Wydall nor Tanaka teach, disclose, or suggest, "replacing a parameter of the step information with a value of the parameter by executing steps specified by the script," as discussed above with respect to the rejection of claim 1. Claim 17 is thus submitted to be allowable as well, for at least those reasons discussed above with respect to the rejection of claim 1. Withdrawal of the rejection of claim 17 is earnestly solicited.

Application Serial No. 10/630,714  
Amendment filed April 14, 2008  
Reply to Office Action mailed November 14, 2007

**Allowable Subject Matter:**

Since no specific grounds of rejection were lodged against claims 3, 7, 11, and 15, acknowledgement of the allowability of claim 3, 7, 11, and 15 is presumed.

**Conclusion:**

Accordingly, in view of the reasons given above, it is submitted that all of claims 1-17 are allowable over the cited references. Allowance of all claims 1-17 and of this entire application is therefore respectfully requested.

Finally, if there are any formal matters remaining after this response, the Examiner is requested to telephone the undersigned to attend to these matters.

If there are any additional fees associated with filing of this Amendment, please charge the same to our Deposit Account No. 19-3935.

Respectfully submitted,

STAAS & HALSEY LLP

Date: April 14, 2008

By: /Thomas E. McKiernan/  
Thomas E. McKiernan  
Registration No. 37,889

1201 New York Avenue, N.W., 7th Floor  
Washington, D.C. 20005  
Telephone: (202) 434-1500  
Facsimile: (202) 434-1501